

# How to Graph the Transmissibility of a Single Degree of Freedom Vibrational System in MATLAB

## Requirements:

- PC with MATLAB installed (Version 2010b was used for this guide)
- Basic understanding of how programming in MATLAB works
- Basic understanding of the free response of a SDOF vibrational system

## List of Steps:

1. Define the System
2. Create the Magnification Curve
3. Create the Phase Lag Curve
4. Input Forcing Function and Calculate the Transmissibility
5. Calculate and Plot the Output Response
6. Interpret the Plots

## Step 1 - Define the System

Start a new script file. If the damping ratio (**zeta**) and natural frequency (**wn**) are known then input their values into the script. If the mass (**m**), damping (**c**), and stiffness (**k**) coefficients are known instead then use these coefficients to calculate the damping ratio and frequency. With these two values the system will completely defined.

---

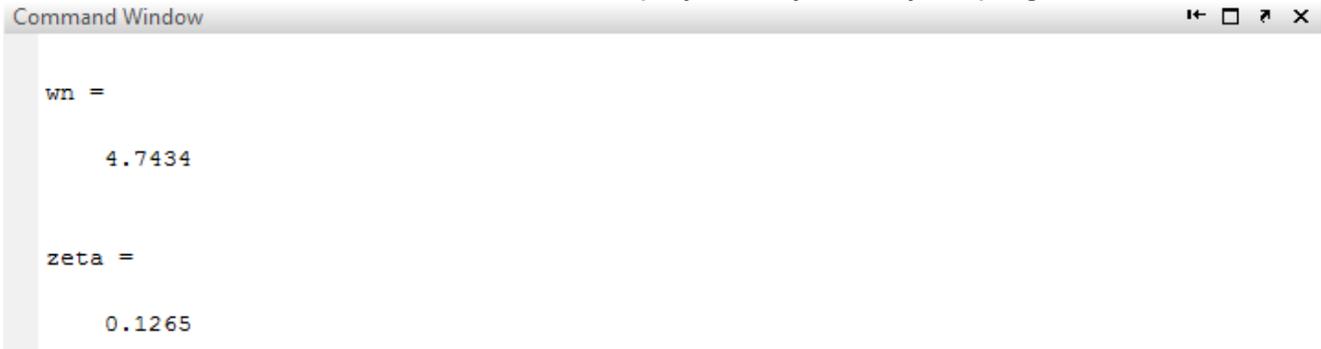
```
%Code for Step 1
%the values below represent example values for an underdamped system that will be
%used as the system throughout this guide
```

```
m = 2000;
c = 2400;
k = 45000;
```

```
wn = sqrt(k/m)
zeta = c/(m*wn*2)
```

---

This is what the command window should display when you run your program:



```
Command Window
wn =
    4.7434

zeta =
    0.1265
```

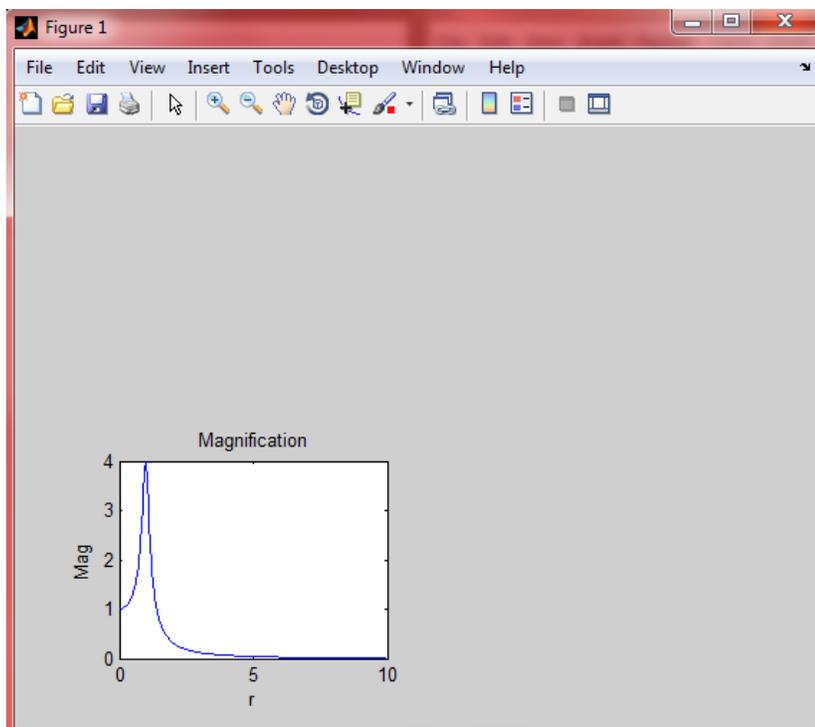
## Step 2 – Create the Magnification Curve

Create a new function named `Magnify.m` with inputs of (`omega`, `wn`, `zeta`). Create a new variable for the range of omega input frequencies to be graphed (`orange`) which will span from 0 to  $10 \cdot wn$  at increments of 0.1. Now create a new variable (`rrange`) which represents the range of the ratios of input frequency to natural frequency. Use the transmissibility of magnification equation to solve for the magnification (`magplot`) over the range of r values. Then have the function plot `magplot` vs `rrange`.

```
%Code for Step 2
function [magnification] = Magnify(omega,wn,zeta)
    r = omega./wn;
    magnification = 1./ (sqrt(((1-r.^2).^2)+(2*zeta.*r).^2));
    orange = 0:.1:wn*10;
    rrange = orange./wn;
    magplot = 1./ (sqrt(((1-rrange.^2).^2)+(2*zeta.*rrange).^2));

    figure(1)
    subplot(2,2,3)
    plot(rrange,magplot)
    title('Magnification')
    xlabel('r')
    ylabel('Mag')
    hold on
end
```

Now add this line to the end of the script file and run the file:  
`Magnify(0,wn,zeta);`



## Step 3 – Create the Phase Lag Curve

Create a new function named `Phase.m` with inputs of `(omega, wn, zeta)`. The phase shift (`phaseplot`) from transmissibility is calculated using an inverse tangent function of `zeta` and `r_angle`. It is physically impossible for a phase shift to be positive, this only occurs due to the nature of the tangent function. If any phase shift is greater than zero then subtract pi from the positive phase shift. Then graph `phaseplot` vs. `r_angle`.

```
%Code for Step 3
function [phaseshift] = Phase(omega,wn,zeta)
    r = omega./wn;
    phaseshift = -atan((2*zeta.*r)./(1-r.^2));

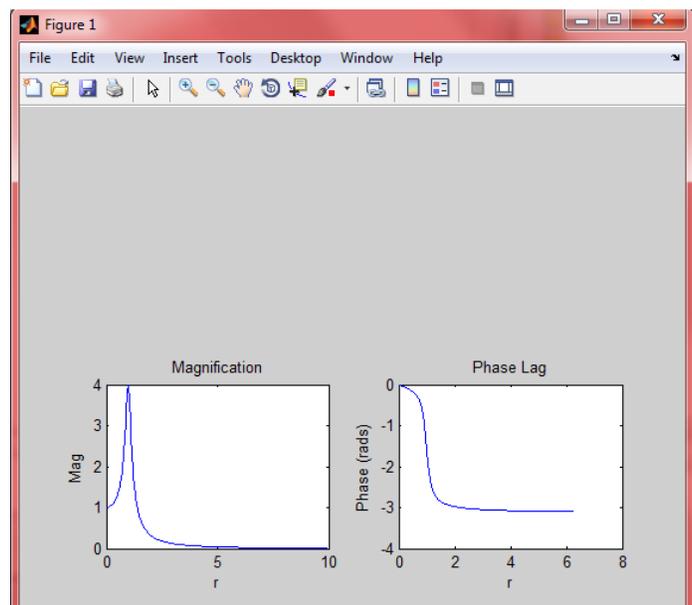
    for p = 1:length(phaseshift)
        if phaseshift(p) > 0
            phaseshift(p) = phaseshift(p) - (pi);
        end
    end

    r_angle = 0:(2*pi)/720:2*pi;
    phaseplot = -atan((2*zeta.*r_angle)./(1-r_angle.^2));
    for n = 1:721
        if phaseplot(n) > 0
            phaseplot(n) = phaseplot(n) - (pi);
        end
    end

    figure(1)
    subplot(2,2,4)
    plot(r_angle,phaseplot)
    title('Phase Lag')
    xlabel('r')
    ylabel('Phase (rads)')
    hold on
end
```

Now add this line to end of the script file and run the script file:

```
Phase(0,wn,zeta);
```



## Step 4 – Input Forcing Function and Calculate the Transmissibility

Add additional lines to the script to input the sine wave forcing function in the following format:

```
force = [A1, A2, A3...]  
omega = [f1, f2, f3...]
```

where  $t$  = time and the input function is expressed as:

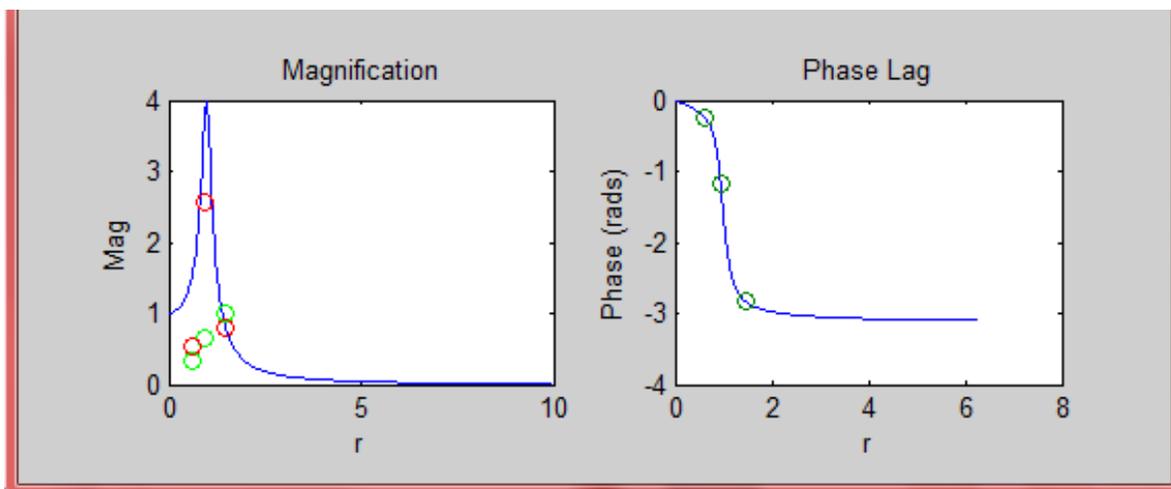
```
[ A1 * sin(f1*t) ] + [ A2 * sin(f2*t) ] + [ A3 * sin(f3*t) ] + ...
```

Then call the function Magnify and Phase and input the **omega**, **wn**, and **zeta**. Calculate the new magnified amplitudes by solving for the displacement at steady state (**qst**) and multiplying by the magnification factors (**mag**). Use a scatter plot to display both the input amplitudes and the magnified amplitudes (**Amag**) as well as the phase shifts.

---

```
%Code for Step 4  
force = [1, 2, 3];  
omega = [3, 4.5, 7];  
  
mag = Magnify(omega,wn,zeta);  
phaseshift = Phase(omega,wn,zeta);  
  
qst = force./k;  
Amag = qst.*mag;  
  
mult = max(qst);  
  
figure(1)  
subplot(2,2,3)  
hold on  
scatter(r,qst./mult,'g')  
scatter(r,Amag./mult,'r')  
  
subplot(2,2,4)  
scatter(r,phaseshift)
```

---



## Step 5 – Calculate and Plot the Output Response

Create a new function `Graphvib` that will take inputs of (`qst, Amag, omega, phaseshifts`) and take these components to create the displacement of the system as a function of time.

Determine the number of input forces and run a for loop to sum the components of displacement for both the input signal (`y`) and the magnified and phase shifted output signal (`ymag`). Then plot the input and output signal vs time.

---

```
%Code for Step 5
function [] = Graphvib(qst,Amag,omega,phaseshift)
maxT = 1/min(omega);
minT = 1/max(omega);

t = 0:minT/2:20*maxT;
x = length(qst);

y = 0;
ymag = 0;

for n = 1:x
    y = y + qst(n)*cos(omega(n).*t);
    ymag = ymag + Amag(n)*cos(omega(n).*t+phaseshift(n));
end

figure(1)
subplot(2,2,1:2)
plot(t,y,'g',t,ymag,'r')
title('Amp vs. t')
xlabel('t')
ylabel('Amp')
hold on

end
```

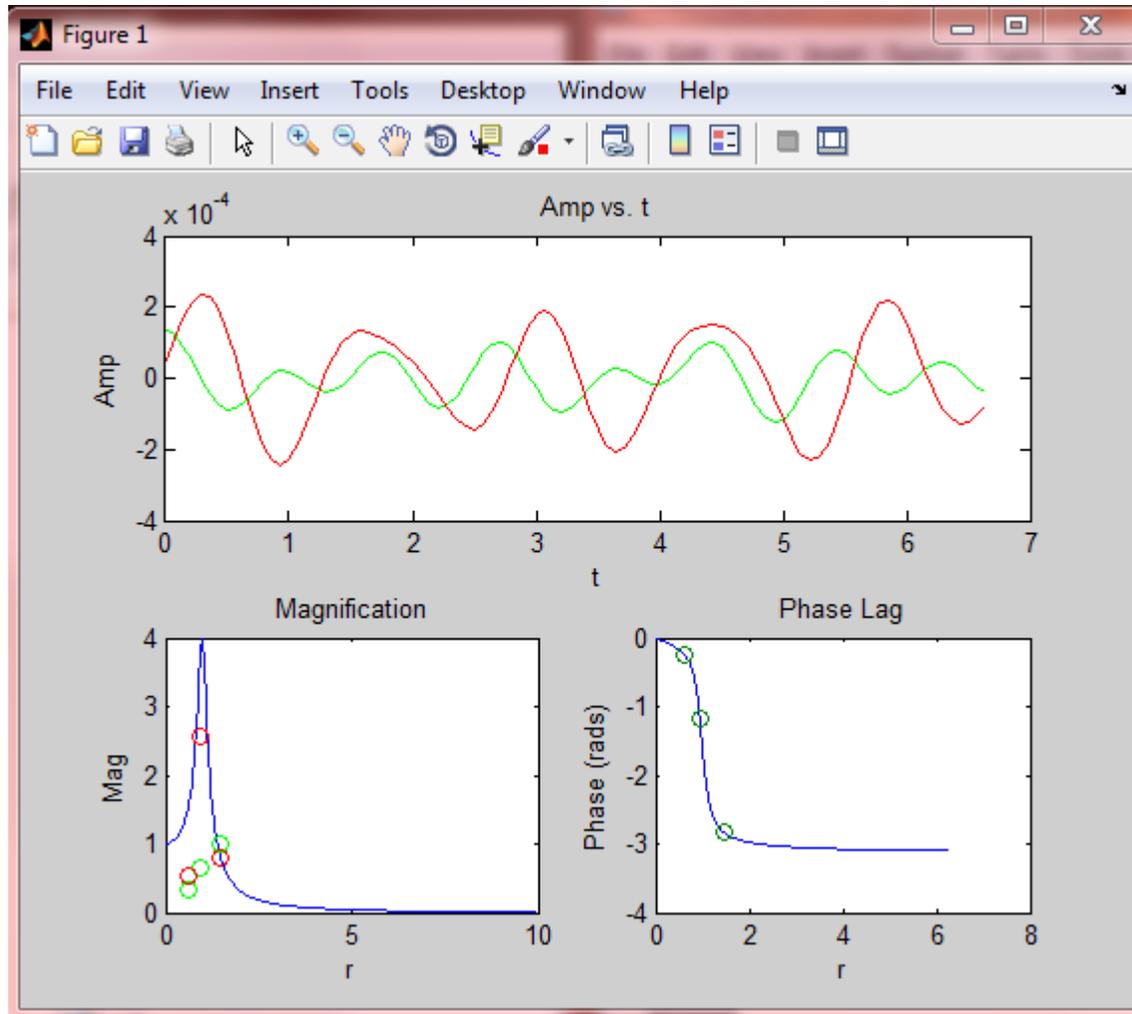
---

Now call this function at the end of the script file:

```
Graphvib(qst,Amag,omega,phaseshift);
```

## Step 6 – Interpret the Plots

Now that all of the programming is finished the only remaining step is to interpret the information given from the plots. The final plot should be:



In both the magnification graph and the amplitude vs. time graph the color green represents the input function and the color red represents the output function. In the magnification graph the height of the dots relative to the blue line is meaningless, but the height of the green dot relative to its paired red dot shows the amount of magnification. A red dot above a green dot shows how much the input signal was amplified, whereas a red dot below a green dot shows how much the input signal was attenuated. It can also be seen how strong each magnified force component is in relation to other magnified force components by comparing each red/green pair of dots. The phase lag plot simply shows how much lag from 0 to  $\pi$  occurs at each input frequency. Remember the  $r$  value is the ratio of the input/output frequency to the natural frequency. Therefore an  $r$  value of 1 is the natural frequency and an  $r$  value greater than 1 is a higher frequency than the natural frequency.